

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Claims 1-64. (cancelled)

Please amend the claims submitted in the preliminary amendment as follows:

65. (currently amended) A computer program product, disposed on a computer-readable storage medium, the product including instructions for causing a processor to:

access Transmission Control Protocol (TCP) segments of a bidirectional TCP connection between a first TCP end-point operating at a first network device and a second TCP end-point operating at a second remote network device;

determine a first TCP state machine state of the first TCP end-point based on at least some of the accessed TCP segments;

determine a second TCP state machine state of the second TCP end-point based on at least some of the accessed TCP segments;

reassemble a first TCP data stream of data sent from the first TCP end-point to the second TCP end-point from payloads of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point based on sequence numbers of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point, at least some of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point being received out of order; and

reassemble a second TCP data stream of data sent from the second TCP end-point to the first TCP end-point from payloads of the accessed TCP segments sent from the second TCP end-point to the first TCP end-point based on sequence numbers of the accessed TCP segments sent from the second TCP end-point to the first TCP end-

point, at least some of the accessed TCP segments sent from the second TCP end-point to the first TCP end-point being received out of order.

66. (previously presented) The computer program product of claim 65, wherein the instructions that determine the TCP state machine state of the first TCP end-point comprise instructions that determine a change in the TCP state machine state of the first TCP end-point.

67. (previously presented) The computer program product of claim 65, wherein the instructions comprise instructions of a software library.

68. (previously presented) The computer program product of claim 67, wherein the instructions comprise instructions of at least one object-oriented class.

69. (previously presented) The computer program of claim 68, wherein the at least one object-oriented class comprises at least one of the following: a class for a bidirectional connection, a class for a TCP end-point, and a class for TCP segment reassembly.

70. (previously presented) The computer program product of claim 65, wherein the instructions to reassemble the first TCP data stream of data sent from the first TCP end-point to the second TCP end-point from the payloads of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point comprise instructions that maintain a linked list storing the first TCP data stream in discontiguous memory locations linked by the linked list.

71. (previously presented) The computer program product of claim 65, wherein the instructions further comprise instructions to provide a return code indicating at least

one of: whether a TCP segment was received out-of-order and whether a TCP segment overlapped another TCP segment.

72. (previously presented) The computer program product of claim 65, wherein the instructions further comprise instructions to return data of a field within a header of a TCP segment.

73. (previously presented) The computer program product of claim 65, further comprising application program instructions that invoke the instructions to access TCP segments, determine the first TCP state machine state, determine the second TCP state machine state, reassemble the first TCP data stream, and reassemble the second TCP data stream.

74. (currently amended) A method, comprising:
providing a software library embodied on a computer readable storage medium
featuring operations to:

access Transmission Control Protocol (TCP) segments of a bidirectional TCP connection between a first TCP end-point operating at a first network device and a second TCP end-point operating at a second remote network device;

determine a first TCP state machine state of the first TCP end-point based on at least some of the accessed TCP segments;

determine a second TCP state machine state of the second TCP end-point based on at least some of the accessed TCP segments;

reassemble a first TCP data stream of data sent from the first TCP end-point to the second TCP end-point from payloads of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point based on sequence numbers of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point, at least some of the accessed TCP segments sent from the first TCP end-point to the second TCP end-point being received out of order; and

reassemble a second TCP data stream of data sent from the second TCP end-point to the first TCP end-point from payloads of the accessed TCP segments sent from the second TCP end-point to the first TCP end-point based on sequence numbers of the accessed TCP segments sent from the second TCP end-point to the first TCP end-point, at least some of the accessed TCP segments sent from the second TCP end-point to the first TCP end-point being received out of order.

75. (previously presented) The method of claim 74, wherein the operations to determine the TCP state machine state of the first end-point comprises operations to determine a change in the TCP state machine state of the first end-point.

76. (previously presented) The method of claim 74, wherein the software library comprises at least one object-oriented class.

77. (previously presented) The method of claim 74, wherein the at least one object-oriented class comprises at least one of the following: a class for a bidirectional connection, a class for a TCP end-point, and a class for TCP segment reassembly.

78. (previously presented) The method of claim 74, wherein the operations to reassemble data from accessed TCP segments sent from the first TCP end-point to the second TCP end-point comprise operations that maintain a linked list storing the first TCP data stream in discontinuous memory locations linked by the linked list.

79. (previously presented) The method of claim 74, wherein the operations further comprise operations to provide a return code indicating at least one of: whether a TCP segment was received out-of-order and whether a TCP segment overlapped another TCP segment.

80. (previously presented) The method of claim 74, wherein the operations further comprise operations to return data of a field within a header of a TCP segment.

81. (previously presented) The method of claim 74, further comprising invoking the software library operations from an application program.